Didier Fallas - InterNexo
difaro@internexo.com

Guía rápida para la generación de un certificado digital

Guía para generar localmente un ente emisor de certificados y que el ente genere un certificado digital para uso de un sitio web.

Generación del ente certificador

1. Crear un directorio de trabajo /etc/apache2/ssl-local

```
# cd /etc/apache2/

# mkdir ssl-local

# cd ssl-local

# mkdir certificados privado
```

2. Generamos unos archivos de control para el ente certificador

En general ahí se lleva control de los certificados generados.

```
# echo '01' > serial

# touch index.txt
```

3. Copiamos el archivo openssl.cnf a /etc/apache2/ssl-local

El contenido del archivo se encuentra al final de este archivo.

4. Generando los datos del ente certificador

```
# cd /etc/apache2/ssl-local

# openssl req -new -x509 -extensions v3_ca -keyout privado/cakey.pem -out cacert.pem
-days 3650 -config ./openssl.cnf
```

Creando un certificado para un dominio

1. Creando directorio de trabajo

```
# cd /etc/apache2/ssl-local

# mkdir certificados/midominio.com
```

2. Creando el CSR (Certificate Signing Request)

```
# openssl req -new -nodes -out certificados/midominio.com/midominio.com.csr -
config ./openssl.cnf
```

3. Creando llaves

```
# mv key.pem certificados/midominio.com/midominio.com.key

# openssl rsa -in certificados/midominio.com/midominio.com.key -out certificados/
midominio.com/midominio.com.key-unenc
```

4. Creando el certificado

```
# openssl ca -out certificados/midominio.com/midominio.com.CERT -config ./openssl.cnf
-days 3650 -infiles certificados/midominio.com/midominio.com.csr
```

Habilitando el certificado en Apache

1. Asegurarse que el módulo ssl esté activo en Apache.

1. Incluir el certificado en la configuración de Apache

Básicamente con las siguientes directivas:

```
SSLEngine on
SSLCipherSuite ALL
SSLCertificateFile /etc/apache2/ssl-local/certificados/midominio.com/
midominio.com.CERT
SSLCertificateKeyFile /etc/apache2/ssl-local/certificados/midominio.com/
midominio.com.key-unenc
```

Archivo openssl.cnf

```
# ***** openssl.cnf ******

dir         = .    # variable que establece el directorio de trabajo

# seccion que permite convertirnos en una CA
# solo se hace referncia a otra sección CA_default
[ ca ]
default_ca    = CA_default

[ CA_default ]
serial        = $dir/serial         # archivo que guarda el siguiente número de serie
database      = $dir/index.txt      # archvio que guarda la bd de certificados
new_certs_dir = $dir/certificados   # dir que guarda los certificados generados
certificate   = $dir/cacert.pem     # nombre del archivo del certificado raíz
```

```
private_key   = $dir/privado/cakey.pem # llave privada del certificado raíz
default_md    = md5                # algoritmo de dispersión usado
preserve      = no                 # Indica si se preserva o no el orden de los
                                   #   campos del DN cuando se pasa a los certs.
nameopt       = default_ca         # esta opcion y la siguiente permiten mostrar
                                   #   detalles del certificado
certopt       = default_ca
policy        = policy_match       # indica el nombre de la seccion
                                   #   donde se especifica que campos son
                                   #   obligatorios, opcionales y cuales deben ser
                                   #   iguales al certificado raíz

# seccion de politicas para la emision de certificados
[ policy_match ]
# Valores: optional, supplied, match
countryName               = supplied         # match, obligatorio
stateOrProvinceName       = supplied
organizationName          = supplied
organizationalUnitName    = supplied      # optional, campo opcional
commonName                = supplied      # supplied, debe estar en la petición
emailAddress              = supplied

# seccion que indica como los certificados deben ser creados
[ req ]
default_bits       = 2048          # tamaño de la llave, si no se indica 512
default_keyfile    = key.pem       # nombre de la llave privada
default_md         = md5           # algoritmo de dispersión a utilizar
string_mask        = nombstr       # caracteres permitidos en la mascara de la llave
distinguished_name = req_distinguished_name  # seccion para el nombre distinguido (DN)
req_extensions     = v3_req        # seccion con mas extensiones que se añaden a la
                                   #   peticion del certificado

# seccion del nombre distinguido, el valor es el prompt que se vera en pantalla.
# datos del propietario del certificado.
# esta seccion define el contenido de datos de id que el certificado llevara.
[ req_distinguished_name ]
0.organizationName         = Nombre de la organizacion
0.organizationName_default  = Organizacion
organizationalUnitName     = Departamento o division
emailAddress               = Correo electronico
emailAddress_max           = 40
localityName               = Ciudad o distrito
localityName_default       = Curridabat
stateOrProvinceName        = Estado o provincia
stateOrProvinceName_default = San Jose
countryName                = Codigo del pais (dos letras)
countryName_default        = CR
countryName_min            = 2
countryName_max            = 2
```

```
commonName                = Nombre comun (hostname o IP)
commonName_max            = 64

# si en la linea de comandos se indica la opcion -x509,
# las siguientes extensiones tambien aplican
[ v3_ca ]
# indica que se trata de un certificado CA raíz con autoridad para
# firmar o revocar otros certificados
basicConstraints       = CA:TRUE

# especifica bajo que metodo identificar a la llave publica que sera certificada
subjectKeyIdentifier   = hash

# especifica como identifcar la    llave publica
authorityKeyIdentifier = keyid:always,issuer:always

# extensiones de la opcion req
[ v3_req ]
basicConstraints          = CA:FALSE  # los certificados firmados no son CA
subjectKeyIdentifier       = hash
# ***********************************************************************************
```